UNITED STATES PATENT AND TRADEMARK OFFICE

A

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/470,329	12/22/1999	BRIAN R. BENNETT	884.174US1	6018

21186 7590 09/26/2005

SCHWEGMAN, LUNDBERG, WOESSNER & KLUTH
P.O. BOX 2938
MINNEAPOLIS, MN 55402-0938

EXAMINER	
TRAN, DENISE	

ART UNIT	PAPER NUMBER
2189	

DATE MAILED: 09/26/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

·X

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

MAILED

SEP 26 2005

Technology Center 2100

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Application Number: 09/470,329
Filing Date: December 22, 1999
Appellant(s): BENNETT ET AL.

Ann M. McCrackin
For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed 4/25/05.

(1) Real Party in Interest

A statement identifying by name the real party in interest is contained in the brief.

(2) Related Appeals and Interferences

RJ

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

(3) Status of Claims

The statement of the status of claims contained in the brief is correct.

(4) Status of Amendments After Final

No amendment after final has been filed.

(5) Summary of Claimed Subject Matter

The summary of claimed subject matter contained in the brief is correct.

(6) Grounds of Rejection to be Reviewed on Appeal

The appellant's statement of the grounds of rejection to be reviewed on appeal is correct.

(7) Claims Appendix

The copy of the appealed claims contained in the Appendix to the brief is correct.

(8) Evidence Relied Upon

6041376	Gilbert et al.	3-2000
6,138,218	Arimilli et al.	2-1998
5,761,446	Donley et al.	6-1998
5,897,656	Vogt et al.	4-1999

(9) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims:

1. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

2. Claims 1-2, 4-5 and 7-8 are rejected under 35 U.S.C. 103(a) as being unpatentable over Gilbert et al., U.S. Patent No. 6,041,376 (hereinafter Gilbert) in view of Arimilli et al., U.S. Patent No. 6,138,218 (hereinafter Arimilli).

As per claim 1, Gilbert shows a method in multiprocessor system (e.g. figs. 1-2 and 6-8C), the method comprising:

identifying a first bus transaction that attempts to modify a shared resource (e.g. fig. 7, el. 76, col. 7, line 5 and col. 9, lines 48-52);

setting a status flag to indicate that a bus transaction attempting to modify the shared resource is pending (e.g., fig. 8C, el. 110; and col. 9, line 63-65 and col. 11, lines 9-20); and

retrying each subsequent nonmodifying bus transaction for the shared resource until the status flag is cleared (e.g. fig. 8C, els. 114-118 and 122; and col. 11, lines 9-20 and col. 9, lines 14-18).

Gilbert does not specifically show the use of the status flag as a bit and preventing live-lock. Arimilli shows the use of a flag as a bit (e.g. col. 3, lines 8-12; and figure 3, element 318 and col. 8, lines 36-44) and preventing live-lock (e.g., col. 2, lines 43-50 and col. 3, lines 1-3). It would have been obvious to one of ordinary skill in the art

at the time the invention was made to apply the teaching of Arimilli to the teaching of Gilbert because it would provide for the storage of the flag (i.e., maintaining the integrity of the flag), minimizing the storage requirements of the system by using a bit, especially when the flag is used to show the use of one state or the other (i.e., two states) and allow other traffic to proceed and alleviate the prospect of a live-lock as taught by Arimilli, col. 3, lines 1-3.

As per claim 7, Gilbert shows the use of a method in a multiprocessor system, the method comprising (e.g. figures 1-2 and 6-8C):

issuing a first bus transaction that attempts to modify a cache line (fig. 6, el. 50; fig. 7, el. 76; and col. 7, line 5 and col. 9, lines 48-52);

setting a status flag to indicate that a bus transaction attempting to modify the cache line is pending (e.g., fig. 8C, el. 110; and col. 9, line 63-65 and col. 11, lines 9-20)

issuing a second bus transaction to read the cache line (e.g., fig. 8C, el. 112 and col. 9, lines 16-18);

retrying the second bus transaction if the status flag is set (e.g. fig. 8C, els. 114-116));

reissuing the first bus transaction that attempts to modify the cache line (e.g., fig. 8C, el. 120 and col. 11, lines 20);

granting the cache line for the reissued first bus transaction if the status flag is set for the cache line (e.g., fig. 8C, el. 122 and col. 11, lines 9-21).

Gilbert does not specifically show the use of the status flag as a bit and preventing live-lock. Arimilli shows the use of a flag as a bit (e.g. col. 3, lines 8-12; and figure 3, element 318 and col. 8, lines 36-44) and preventing live-lock (e.g., col. 2, lines 43-50 and col. 3, lines 1-3). It would have been obvious to one of ordinary skill in the art at the time the invention was made to apply the teaching of Arimilli to the teaching of Gilbert because it would provide for the storage of the flag (i.e., maintaining the integrity of the flag), minimizing the storage requirements of the system by using a bit, especially when the flag is used to show the use of one state or the other (i.e., two states) and allow other traffic to proceed and alleviate the prospect of a live-lock as taught by Arimilli, col. 3, lines 1-3.

As per claims 2 and 8, Gilbert teaches clearing the status flag when the reissued first bus transaction completes (e.g., fig. 8C, el. 122). As stated above, Gilbert does not specifically show the use of the status flag as a bit. Arimilli shows the use of a flag as a bit (e.g. col. 3, lines 8-12; and figure 3, element 318 and col. 8, lines 36-44). It would have been obvious to one of ordinary skill in the art at the time the invention was made to apply the teaching of Arimilli to the teaching of Gilbert because it would provide for the storage of the flag (i.e., maintaining the integrity of the flag), minimizing the storage requirements of the system by using a bit, especially when the flag is used to show the use of one state or the other (i.e., two states) and allow other traffic to proceed and alleviate the prospect of a live-lock as taught by Arimilli col. 3, lines 1-3.

As per claim 4, Gilbert shows the use of clearing the status flag at periodic intervals (e.g., fig. 8C, el. 124). As stated above, Gilbert does not specifically show the use of the status flag as a bit. Arimilli shows the use of a flag as a bit (e.g. col. 3, lines 8-12; and figure 3, element 318 and col. 8, lines 36-44). It would have been obvious to one of ordinary skill in the art at the time the invention was made to apply the teaching of Arimilli to the teaching of Gilbert because it would provide for the storage of the flag (i.e., maintaining the integrity of the flag), minimizing the storage requirements of the system by using a bit, especially when the flag is used to show the use of one state or the other (i.e., two states) and allow other traffic to proceed and alleviate the prospect of a live-lock as taught by Arimilli, col. 3, lines 1-3.

As per claim 5, Gilbert shows the use of clearing the status flag at periodic intervals (e.g., fig. 8C, el. 124) and a given period of time can be any desired value (e.g., col. 11, lines 29-44). As stated above, Gilbert does not specifically show the use of the status flag as a bit. Arimilli shows the use of a flag as a bit (e.g. col. 3, lines 8-12; and figure 3, element 318 and col. 8, lines 36-44. It would have been obvious to one of ordinary skill in the art at the time the invention was made to apply the teaching of Arimilli to the teaching of Gilbert because it would provide for the storage of the flag (i.e., maintaining the integrity of the flag), minimizing the storage requirements of the system by using a bit, especially when the flag is used to show the use of one state or the other (i.e., two states) and allow other traffic to proceed and alleviate the prospect of a live-lock as taught by Arimilli, col. 3, lines 1-3.

Gilbert does not explicitly show periodical intervals being longer than a length of time for a bus transaction to complete. "Official Notice" is taken that both the concept and advantage of having periodical intervals being longer than a length of time for a bus transaction to complete are well known and expected in the art. It would have been obvious to one of ordinary skill in the art at the time the invention was made to have the periodical intervals are longer than a length of time for a bus transaction to complete because it would allow sufficient time to finish the transaction and reduce the number of retry requests, thereby improving the system performance.

3. Claims 3, 6, and 9 are rejected under 35 U.S.C. 103(a) as being unpatentable over Gilbert et al., U.S. Patent No. 6,041,376, (hereinafter Gilbert) in view of Arimilli et al., U.S. Patent No. 6,138,218 (hereinafter Arimilli), as applied to claims 1 and 7 above, and further in view of Donley et al., U.S. Patent No. 5,761,446 (hereinafter Donley).

As per claims 3, 6, and 9, Gilbert shows the use of clearing the status flag by a counter where the counter can be any desired value (e.g., col. 11, lines 29-44). As stated above, Gilbert does not specifically show the use of the status flag as a bit. Arimilli shows the use of a flag as a bit (e.g. col. 3, lines 8-12; and figure 3, element 318 and col. 8, lines 36-44). It would have been obvious to one of ordinary skill in the art at the time the invention was made to apply the teaching of Arimilli to the teaching of Gilbert because it would provide for the storage of the flag (i.e., maintaining the integrity of the flag), minimizing the storage requirements of the system by using a bit, especially when the flag is used to show the use of one state or the other (i.e., two states) and

allow other traffic to proceed and alleviate the prospect of a live-lock as taught by Arimilli, col. 3, lines 1-3.

Gilbert and Arimilli do not specifically show the use of randomly or pseudo-randomly. Donley shows generating random number or Pseudo- random number (e.g., col. 3, lines 46-60). It would have been obvious to one of ordinary skill in the art, at the time the invention was made, to apply the teaching of Donley to the combine system of Gilbert and Arimilli because it would provide a random delay time, thereby optimizing live-lock avoidance and system performance as taught by Donley, col. 2, lines 61-63.

4. Claims 10-21 are rejected under 35 U.S.C. 103(a) as being unpatentable over Vogt et al., U.S. Patent No. 5,897,656, (hereinafter Vogt) in view of Gilbert et al., U.S. Patent No. 6,041,376) (hereinafter Gilbert).

As per claim 10, Vogt shows the use of a multiprocessor computer system comprising:

a plurality of processors (e.g., fig. 1, els. 112);

a resource shared by the plurality of processors (e.g., fig.1, el. 132);

at least one system bus interconnecting the shared resource and the plurality of processors (e.g., fig.1, el. 102);

a plurality of buffers, each one of the plurality of buffers associated with a bus transaction initiated on the at least one system bus by one of the processors (e.g. figure 2, elements 208 and 210 and figure 5A, els 500 and fig. 6; and col. 25, line 54 to col. 26, line 27); and

a status indicator associated with each of the plurality of buffers (e.g. col. 26, lines 16-27; col. 28, lines 24-26), and a first one of the processors initiates a bus transaction attempting to modify the shared resource and the bus transaction is retried (e.g., col. 25, lines 61-64; and col. 27, lines 32-45).

Vogt does not specifically show the status indicator to indicate when a first one of the processors initiates a bus transaction attempting to modify the shared resource. Gilbert shows the use of a status indicator to indicate when a first one of the processors initiates a bus transaction attempting to modify the shared resource (e.g., fig. 8C, el. 110; col. 9, line 63-65 and col. 11, lines 9-20; and fig. 7, el. 76; col. 7, line 5 and col. 9, lines 48-52). It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine Gilbert with Vogt because it would provide guaranteed forward progress of processor requests for data by preventing other processors from accessing data until the processor request is satisfied as taught by Gilbert col.2, lines 41-50.

As per claim 11, Vogt shows the use of four processors are coupled to each one of the system buses (e.g. figure 1, elements 112a-d and additional processors interpreted as at least four; and col. 16, lines 50-57).

As per claim 12, Vogt shows the use of at least one system bus comprises two processor buses (e.g. figure 1, elements 102 and 104).

As per claim 13, Vogt shows the use of four processors coupled to each one of the two processor buses (e.g. figure 1, elements 102, 104 and 112a-d and additional processors interpreted as at least four; and col. 16, lines 50-57.

As per claim 14, Vogt shows the use of an input/output bus (e.g. figure 1, element 106).

As per claim 15, Vogt shows the use of a multiple bus, multiprocessor computer system (e.g., fig. 1, els 102, 104, and 112) comprising:

a plurality of processors (fig. 1, els. 112);

a plurality of data cache memories (e.g. fig. 1, els. 114);

a system memory shared by the plurality of processors (e.g. fig. 1, el. 132);

at least two buses interconnecting the system memory with the plurality of data cache memories and the plurality of processors (e.g. fig. 1, els. 102 and 104); and

a controller (e.g. fig. 1, el. 130) comprising:

a plurality of buffers, each one of the plurality of buffers associated with a bus transaction initiated on one of the buses by one of the processors (e.g. fig. 2, els. 208 and 210 and fig. 5A, els. 500 and fig. 6; and col. 25, line 54 to col. 26, line 27); and

a status indicator associated with each of the plurality of buffers (e.g. col. 26, lines 16-27; col. 28, lines 24-26), and a first one of the processors initiates a bus transaction attempting to modify the system memory and the bus transaction is retried (e.g., col. 25, lines 61-64; and col. 27, lines 32-45).

Vogt does not specifically show the status indicator to indicate when a first one of the processors initiates a bus transaction attempting to modify the shared resource. Gilbert shows the use of a status indicator to indicate when a first one of the processors initiates a bus transaction attempting to modify the system memory (e.g., fig. 8C, el. 110; col. 9, line 63-65 and col. 11, lines 9-20; and fig. 7, el. 76; col. 7, line 5 and col. 9, lines 48-52). It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine Gilbert with Vogt because it would provide guaranteed forward progress of processor requests for data by preventing other processors from accessing data until the processor request is satisfied as taught by Gilbert col.2, lines 41-50.

As per claim 16, Vogt shows each one of the at least two buses is coupled to four of the processors (e.g. figure 1, elements 102, 104 and 112a-d and additional processors interpreted as at least four; and col. 16, lines 50-57).

As per claim 17, Vogt shows the use of an integrated circuit (e.g., fig. 2, el. 130 and col. 18, lines 43-46) comprising:

a bus interface to control a plurality of bus transactions (e.g. fig. 1, el. 204);

a coherency module to maintain cache coherency for a plurality of cache lines (e.g., fig. 2, el. 200); and

a buffer manager (e.g., fig. 2, el. 210) comprising,

a plurality of buffers (e.g. fig. 5A, els. 500), each one of the buffers to store information associated with one of the plurality of bus transactions received by the bus interface (e.g. fig. 5A, els. 500 and col. 26, lines 7-27); and

a plurality of status indicators where at least one of the status indicators associated with each of the buffers (e.g. fig. 5A, els. 502, 505), and one of the bus transactions attempting to modify one of the cache lines is retried (e.g., col. 25, lines 61-64; and col. 27, lines 32-45).

Vogt does not specifically show the status indicators indicating that one of the bus transactions attempting to modify one of the cache lines is retried. Gilbert shows the use of status indicators indicating that one of the bus transactions attempting to modify one of the cache lines is retried (e.g., fig. 8C, els. 110-122; col. 9, line 63-65 and col. 11, lines 9-24; and fig. 7, el. 76; col. 7, line 5 and col. 9, lines 48-52). It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine Gilbert with Vogt because it would provide guaranteed forward progress of processor requests for data by preventing other processors from accessing data until the processor request is satisfied as taught by Gilbert col.2, lines 41-50.

As per claim 18, Vogt shows the use of the buffer manager further comprises logic to determine a type of bus transaction occurring on a bus (e.g. fig. 5A, el. 505 and col. 26, lines 24-27).

As per claim 19, Vogt shows the use of the buffer manager further comprises logic to determine if two of the bus transactions are contending for a same cache line (e.g. fig. 5B, els. 510 and col. 14, lines 10-30, col. 27, line 22 to col. 28, line 16).

As per claim 20, Vogt does not explicitly show logic to reset all of the plurality of status indicators. Gilbert shows logic to reset all of the plurality of status indicators (e.g., fig. 8C, els. 124 or 122; col. 2, lines 51-60; and col. 11, lines 9-30, where all the status flag can be reset at different times). It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine Gilbert with Vogt because it would provide guarantee forward progress of processor requests for data by preventing other processors from accessing data until the processor request is satisfied and by limiting the amount of time that other processors are prevented from accessing the data as taught by Gilbert col. 2, line 41 to col. 3, line 8.

As per claim 21, Vogt shows the use of 64 buffers and 64 status indicators (e.g. figure 5A, els 500 and 502, 505 and col. 26, lines 15-20).

(10) Response to Argument

1. In the remarks, Appellant argued in substance that (1) there is no teaching or suggestion in Gilbert et al. of setting a status bit to indicate an attempt to modify a shared resource.

As to point (1) the examiner respectfully disagrees. As stated in the 103 rejections with respect to claims 1 or 7 above, the combination of Gilbert and Arimilli

teaches setting a status bit to indicate an attempt to modify a shared resource. In particular, Gilbert shows a bus transaction that attempts to modify a shared resource (e.g. fig. 7, el. 76, col. 7, line 5 and col. 9, lines 48-52) and setting a status flag to indicate that a bus transaction attempting to modify the shared resource is pending (e.g., fig. 8C, el. 110; and col. 9, line 63-65 and col. 11, lines 9-20); and Arimilli shows the use of a flag as a bit (e.g. col. 3, lines 8-12; and figure 3, element 318 and col. 8, lines 36-44).

Specifically, in Gilbert figure 8C, element. 110, 120, 122 and col. 11, lines 9-20, Gilbert not only shows "with the hold for forward progress field 82 set, snoopy cache protocol engine 32 can prevent access to all data request from other nodes" but also shows setting the hold flag to indicate a requested data line "being held for a local processor" or being held for a request of the processor until "if the requesting processor retries its request the installed data line is delivered to the requesting processor," col. 11, lines 20-22. In addition, Gilbert teaches the processor request is a write (i.e., modify) request, col. 7, line 5, "if the processor request is a write" and col. 9, lines 18-22 " [A] processor usually seeks to control of a data line to write to the line." Therefore, Gilbert does teach setting a status flag to indicate that a (write request or modify request) bus transaction attempting to modify a data line or shared resource. Thus, the combination of Gilbert and Arimilli teaches setting a status bit to indicate an attempt to modify a shared resource.

2. In the remarks, Appellant argued in substance that (2) the hold flag referred to in

the Office Action is used by Gilbert et al. to indicate that a remote cache has received data, and not to indicate "setting a status bit to indicate that a bus transaction attempting to modify the shared resource is pending," as recited in claim 1, or "setting a status bit to indicate that a bus transaction attempting to modify the cache line is pending," as recited in claim 7, Appeal Brief filed 4/25/05, page 10, last paragraph.

As to point (2) the examiner respectfully disagrees. Gilbert, col. 10, lines 60-65, describes "[After] receiving the data line and if the retry field 78 has been set, the snoopy cache protocol engine sets a hold flag." Thus, the hold flag is set when the retry field has been set. Gilbert also, teaches "[If] the retry field is set, the snoopy cache protocol engine 32 wait for the processor to issued a retry request," col. 9, lines 53-56. Therefore, the hold flag is used to indicate that the snoopy cache protocol engine is waiting for a request bus transaction to be retried. In addition, Gilbert teaches the processor request bus transaction is a write (i.e., modify) request, col. 7, line 5, "if the processor request is a write" and col. 9, lines 18-22 " [A] processor usually seeks to control of a data line to write to the line." Therefore, it is clear that Gilbert teaches the setting of status flag to indicated that a (write request or modify request) bus transaction is attempting to modify a shared resource or a cache line is pending as recited in claims 1 and 7 respectively.

More over, as stated in the 103 rejections with respect to claims 1 or 7 above, the combination of Gilbert and Arimilli teaches setting a status bit to indicate an attempt to modify a shared resource. In particular, Gilbert shows a bus transaction that attempts to modify a shared resource (e.g. fig. 7, el. 76, col. 7, line 5 and col. 9, lines

48-52) and setting a status flag to indicate that a bus transaction attempting to modify the shared resource is pending (e.g., fig. 8C, el. 110; and col. 9, line 63-65 and col. 11, lines 9-20); and Arimilli shows the use of a flag as a bit (e.g. col. 3, lines 8-12; and figure 3, element 318 and col. 8, lines 36-44).

Specifically, in Gilbert figure 8C, element. 110, 120, 122 and col. 11, lines 9-20, Gilbert not only shows "with the hold for forward progress field 82 set, snoopy cache protocol engine 32 can prevent access to all data request from other nodes" but also shows setting the hold flag to indicate a requested data line "being held for a local processor" or being held for a request of the processor until "if the requesting processor retries its request the installed data line is delivered to the requesting processor," col. 11, lines 20-22. In addition, Gilbert teaches the processor request is a write (i.e., modify) request, col. 7, line 5, "if the processor request is a write" and col. 9, lines 18-22 " [A] processor usually seeks to control of a data line to write to the line." Therefore, Gilbert does teach setting a status flag to indicate that a (write request or modify request) bus transaction attempting to modify a data line or shared resource. Thus, the combination of Gilbert and Arimilli teaches setting a status bit to indicate an attempt to modify a shared resource.

3. In the remarks, Appellant argued in substance that (3) the '"in used"' hold flag of Gilbert, which is used to indicate that a remote cache has received data, fails to teach the setting of status bit <u>to indicate that a bus transaction is attempting to modify</u> either a shared resource or a cache line is pending, as recited in claims 1 and 7, Appeal Brief

filed 4/25/05, page 11, lines 25-31.

As to point (3) the examiner respectfully disagrees. Gilbert not only teaches the hold flag set in response to a remote cache has received data but also teaches the requested data is transmitted to the remote cache in response to the processor write request (i.e., modify request) transaction initiated (col. 7, line 5 and lines 45-65). Since the setting of the hold flag when the processor write request transaction is initiated, it is clear that the hold flag is a status flag to indicate that a (write request or modify request) bus transaction attempting to modify a shared resource or a cache line pending as recited in claims.

More over, as stated in the 103 rejection with respect to claims 1 or 7 above, the combination of Gilbert and Arimilli teaches setting a status bit to indicate an attempt to modify a shared resource. Gilbert shows a bus transaction that attempts to modify a shared resource (e.g. fig. 7, el. 76, col. 7, line 5 and col. 9, lines 48-52) and setting a status flag to indicate that a bus transaction attempting to modify the shared resource is pending (e.g., fig. 8C, el. 110; and col. 9, line 63-65 and col. 11, lines 9-20); and Arimilli shows the use of a flag as a bit (e.g. col. 3, lines 8-12; and figure 3, element 318 and col. 8, lines 36-44). Thus, combination of Gilbert and Arimilli teaches setting a status bit to indicate an attempt to modify a shared resource or cache line is pending as recited in claims 1 and 7.

4. In the remarks, Appellant argued in substance that (4) Arimilli fails to teach setting of a status bit to indicate that a bus transaction is attempting to modify either a

shared resource or a cache line.

In response to applicant's arguments against the references individually, one cannot show nonobviousness by attacking references individually where the rejections are based on combinations of references. See *In re Keller*, 642 F.2d 413, 208 USPQ 871 (CCPA 1981); *In re Merck & Co.,* 800 F.2d 1091, 231 USPQ 375 (Fed. Cir. 1986). In this case the combination of Gilbert and Arimilli teaches setting a status bit to indicate an attempt to modify a shared resource. In particular, Gilbert shows a bus transaction that attempts to modify a shared resource (e.g. fig. 7, el. 76, col. 7, line 5 and col. 9, lines 48-52) and setting a status flag to indicate that a bus transaction attempting to modify the shared resource is pending (e.g., fig. 8C, el. 110; and col. 9, line 63-65 and col. 11, lines 9-20); and Arimilli shows the use of a flag as a bit (e.g. col. 3, lines 8-12; and figure 3, element 318 and col. 8, lines 36-44). Thus, combination of Gilbert and Arimilli teaches setting a status bit to indicate an attempt to modify a shared resource or cache line is pending as recited in claims 1 and 7.

5. In the remarks, Appellant argued in substance that (5) the Office Action fails to point to anything of record in either Gilbert et al. or Arimilli et al. that would indicate a finding of a teaching, suggestion, or motivation to combine the references of Gilbert et al. and Arimilli et al. Further, there is no indication in the Office Action of how maintaining the integrity of the flag and minimizing the storage requirements of the system would allow other traffic to proceed or to alleviate the prospect of live-lock.

As to point (5) to begin the examiner would like to point out that the actually

motivational statement the examiner used was "it would provide for the storage of the flag (i.e., maintaining the integrity of the flag) minimizing the storage requirements of the system by using a bit, especially when the flag is used to show the use of one state or the other (i.e., two states) **and** allow other traffic to proceed and alleviate the prospect of a live-lock as taught by Arimilli, col. 3, lines 1-3." Appellant left out "and." The rejection did not state that "maintaining the integrity of the flag and minimizing the storage requirements of the system **would allow** other traffic to proceed or to alleviate the prospect of live-lock, " as appellant stated in the Appeal Brief filed 4/25/05, page 13.

As previous stated in the Non Final rejection mailed 11/16/04, page 21, lines 3-10, the examiner has in fact pointed to the teaching, suggestion, or motivation comes from the knowledge of one skilled in the art and the reference relied upon, such as Arimilli, col. 3, lines 1-3. Also, the teaching, suggestion, or motivation of "maintaining the integrity of the flag, minimizing the storage requirements of the system by using a bit, especially when the flag is used to show the use of one state or the other (i.e., two states)" are come from the knowledge of one skilled in the art. As per "allow other traffic to proceed and alleviate the prospect of a live-lock" Arimilli shows this, col. 3, lines 1-3.

6. In the remarks, Appellant argued in substance that (6) Donley et al., fails to teach or suggest clearing of a status bit randomly or pseudo-randomly as recited in claims 3, 6, and 9.

As to point (6) the examiner respectfully disagrees. As stated in the last office

action, Gilbert shows the use of clearing the status flag by a counter where the counter can be any desired value (e.g. col. 11, lines 29-44), however Gilbert does not show the use of the status flag as a bit. Arimilli shows the use of a flag as a bit (e.g. col. 3, lines 6-12; and figure 3, element 318 and col. 8, lines 36-44). Gilbert and Arimilli do not specifically show the use of randomly or pseudo-randomly. Donley shows generating random number or pseudo-random number (e.g. col. 3, lines 46-60). It is the combination of references relied upon to reject the claimed limitations of claims 3, 6, and 9 not Donley in of itself (piecemeal analysis). Therefore by combining Donley with Gilbert, it would allow for the counter to be set with a random or pseudo-random number in order to optimize live-lock avoidance, as taught by Donley, col. 2, lines 61-63.

7. In the remarks, Appellant argued in substance that (7) the Office Action fails to explain how providing a random delay time would optimize the system in Gilbert and Arimilli and there is no indication of whether there is a reasonable chance of success in combining the random delay time of Donley with Gilbert and Arimilli.

As to point (7) the examiner respectfully disagrees. As stated in the last office action, Gilbert shows the use of clearing the status flag by a counter where the counter can be any desired value (e.g. col. 11, lines 29-44), however Gilbert does not show the use of the status flag as a bit. Arimilli shows the use of a flag as a bit (e.g. col. 3, lines 6-12; and figure 3, element 318 and col. 8, lines 36-44). Gilbert and Arimilli do not specifically show the use of randomly or pseudo-randomly. Donley shows generating random number or pseudo-random number (e.g. col. 3, lines 46-60). Therefore by

combining Donley with Gilbert, it would allow for the counter to be set with a random or pseudo-random number in order to optimize live-lock avoidance, as taught by Donley, col. 2, lines 61-63.

8. In the remarks, Appellant argued in substance that (8) with respect to claims 10, and 15, Gilbert et al. failed to teach or suggest "to indicate when a first one of the processors initiates a bus transaction attempting to modify the shared resource and the bus transaction is retried."

As to point (8) the examiner respectfully disagrees. In response to applicant's arguments against the references individually, one cannot show nonobviousness by attacking references individually where the rejections are based on combinations of references. See *In re Keller*, 642 F.2d 413, 208 USPQ 871 (CCPA 1981); *In re Merck & Co.,* 800 F.2d 1091, 231 USPQ 375 (Fed. Cir. 1986). In this case, the combination of Vogt and Gilbert shows "to indicate when a first one of the processors initiates a bus transaction attempting to modify the shared resource and the bus transaction is retried." In particular, Vogt shows the bus transaction is retried (e.g., col. 25, lines 61-64; and col. 27, lines 32-45). Gilbert shows the use of a status indicator to indicate when a first one of the processors initiates a bus transaction attempting to modify the shared resource (e.g., fig. 8C, el. 110; col. 9, line 63-65 and col. 11, lines 9-20; and fig. 7, el. 76; col. 7, line 5 and col. 9, lines 48-52).

Specifically, Gilbert shows a status bit (i.e., hold flag, col. 9, lines 63-65) to indicate that a first one of the processors initiates a bus transaction attempting to modify

a shared resource or cache line (i.e., a write request to data not stored locally) (e.g. fig. 8C, elements 110-122; col. 9, lines 63-65 and col. 11, lines 9-24; and fig. 7, el. 76; col. 7, line 5 and col. 9, lines 48-65). Therefore the combination of references show "to indicate when a first one of the processors initiates a bus transaction attempting to modify the shared resource and the bus transaction is retried." Furthermore, Gilbert shows the use of the bus transaction is retried (e.g. figure 8C, element 120, 122, col. 11, lines 9-20, col. 9, lines 14-18 and lines 48-65).

In further discussion, Gilbert not only teaches the hold flag set in response to a remote cache has received data but also teaches the requested data is transmitted to the remote cache in response to the processor write request (i.e., modify request) transaction initiated (col. 7, line 5 and lines 45-65). Since the setting of the hold flag when the processor write request transaction initiated, it is clear that the hold flag is a status flag to indicate that a (write request or modify request) bus transaction attempting to modify a shared resource or a cache line pending as recited in claims.

9. In the remarks, Appellant argued in substance that (9) with respect to claim 17, Gilbert et al. failed to teach or suggest "to indicate that one of the bus transactions attempting to modify one of the cache lines is retried."

As to point (9) the examiner respectfully disagrees. Gilbert shows the use of status indicators indicating that one of the bus transactions attempting to modify one of the cache lines is retried (e.g., fig. 8C, els. 110-122; col. 9, line 63-65 and col. 11, lines 9-24; and fig. 7, el. 76; col. 7, line 5 and col. 9, lines 48-52).

10. In the remarks, Appellant argued in substance that (10) the motivation to combine Vogt with Gilbert is merely a conclusory statement based on subjective belief, and further that the statement is not supported by the record.

As to point (10) the examiner respectfully disagrees. In response to applicant's argument that there is no suggestion to combine the references, the examiner recognizes that obviousness can only be established by combining or modifying the teachings of the prior art to produce the claimed invention where there is some teaching, suggestion, or motivation to do so found either in the references themselves or in the knowledge generally available to one of ordinary skill in the art. See *In re Fine*, 837 F.2d 1071, 5 USPQ2d 1596 (Fed. Cir. 1988)and *In re Jones*, 958 F.2d 347, 21 USPQ2d 1941 (Fed. Cir. 1992). In this case, as stated previously, Gilbert (col. 2, lines 41-50) shows the use of the advantage of having status indicators to provide guaranteed forward progress of processor requests for data by preventing other processors from accessing data until the processor request is satisfied. Therefore the motivation is neither a conclusory statement based on subjective belied nor not supported by the record.

11. In the remarks, Appellant argued in substance that (11) since Gilbert makes no disclosure of checking for address conflict, combining Gilbert with Vogt does not guarantee forward progress of processor request.

As to point (11) the examiner respectfully disagrees. Since Gilbert col. 11, lines 14-20, teaches checking for data line conflict, “checks the particular data line requested

by another node and only rejects the request if it is for the same data line being held for a local processor, " combining Gilbert with Vogt does guarantee forward progress of processor request as taught by Gilbert (col. 2, lines 41-50). Hence the statements in the Office Action are supported by the record, and further demonstrate that there is a reasonable expectation of success. Thus the Office action establish a prima facie case of obviousness with respect to claims 10-21.

For at least the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

Denise Tran
PPE of 2189
September 19, 2005

Conferees:
Matthew Kim
SPE of 2186

MATTHEW KIM
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100

DONALD SPARKS
SUPERVISORY PATENT EXAMINER

Donald Spark
SPE of 2187

SCHWEGMAN, LUNDBERG, WOESSNER & KLUTH, P.A.
P.O. BOX 2938
MINNEAPOLIS, MN 55402-0938